Q.1

A substitution cipher is a method of encryption which involves replacing each symbol in the original message with another symbol – ciphers in which the substitution remains unchanged throughout the message are known as *monoalphabetic substitution ciphers*. [1]

Since the substitution pattern stays the same throughout the process of encryption, i.e., α always maps to β, monoalphabetic substitution ciphers can be very vulnerable to statistical analysis. By keeping track of the most frequently occurring symbols within the ciphertext, we can then infer that these symbols have been used to replace the most frequently used symbols in the plaintext's language. Once the most popular symbols have been decrypted, we can deduce the substitution of the other symbols by utilising the redundancy that exists within natural language.

```
['x', 'j', 's', 'd', 'y', 'm', 'p', 'a', 'n', 'e', 't', 'z', 'w', 'h', 'c', 'l', 'r', 'q', 'b', 'u', 'g', 'o', 'k']
[' ', 'e', 't', 'a', 'i', 'o', 'n', 's', 'r', 'h', 'l', 'c', 'd', 'u', 'm', 'p', 'f', 'g', 'y', 'w', 'b', 'v', 'k', 'x',
 'j', 'q', 'z']
wher we fogyane the irlimiluads oc the sage manietp on submanietp oc oun odlen fudtimatel ydarts arl arigads ore oc the
cinst yoirts whifh stnives us is that thep kerenaddp liccen gone cnog eafh othen thar lo the irlimiluads oc arp ore syef
ies on manietp ir a state oc ratune

['x', 'j', 's', 'd', 'y', 'm', 'p', 'a', 'n', 'e', 't', 'z', 'w', 'h', 'c', 'l', 'r', 'q', 'b', 'u', 'g', 'o', 'k']
[' ', 'e', 't', 'a', 'i', 'o', 'r', 's', 'n', 'h', 'l', 'c', 'd', 'u', 'm', 'p', 'f', 'g', 'y', 'w', 'b', 'v', 'k', 'x',
 'j', 'q', 'z']
when we fogyare the inlimiluads oc the sage marietp or submarietp oc our odler fudtimatel ydants anl anigads one oc the
cirst yoints whifh strives us is that thep keneraddp liccer gore crog eafh other than lo the inlimiluads oc anp one syef
ies or marietp in a state oc nature
```

Figure 1: the output of my program illustrating the early stages of decryption. Useful to demonstrate the redundancy in natural language.

For example, in figure 1 the first word in the plaintext is most likely 'when' – other options include 'whey' or 'whet', but they are considerably less likely. In the amended section of plaintext that follows, the text 'one oc the cirst' is visible near the bottom right of figure 1 – this should most likely be 'one of the first' instead. By utilising this natural redundancy, and continually updating the key I was using to attempt to decrypt the cipher, I eventually arrived at the plaintext that follows:

'*when we compare the individuals of the same variety or subvariety of our older cultivated plants and animals one of the first points which strikes us is that they generally differ more from each other than do the individuals of any one species or variety in a state of nature*'.

The process I used to decrypt the cipher is also detailed in the comments of the attached python program named 'decodeCipher.py'.

Q.2

ChaCha20 is a modern stream cipher introduced for use in TLS 1.3 – it works with a 512-bit state, divided into sixteen 32-bit words. These words are altered by three operations: addition mod 2^{32}, XOR, and <<< n (rotate left by n bits).[2] In this question, we were asked to consider a smaller version of the ChaCha20 cipher, in which the state consisted of just 64 bits.

The four basic steps in a quarter round applied to the first column of the initial state can be seen in detail below in figures 2, 3, and 4.

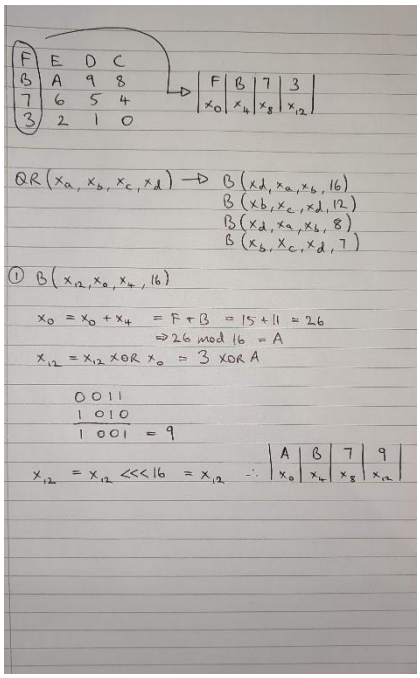

Figure 2: page one of detailed quarter round

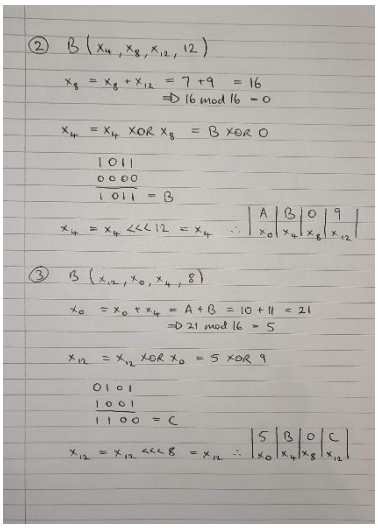

Figure 3: page two of detailed quarter round

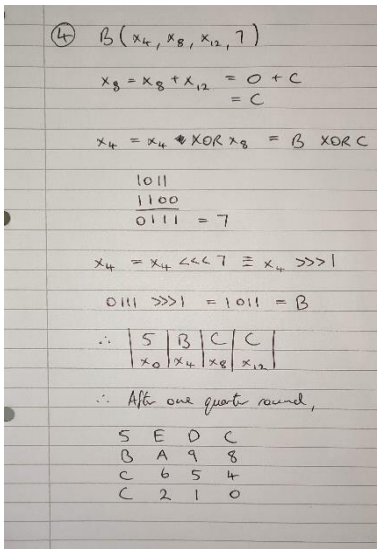

Figure 4: page three of detailed quarter round

After one quarter round applied to the first column, the state looked like this:

5	E	D	C
B	A	9	8
C	6	5	4
C	2	1	0

After four quarter rounds applied to the columns, the state looked like this:

5	2	B	4
B	1	E	4
C	8	8	8
C	8	C	0

The state is colour-coded to indicate the diagonals involved in the diagonal round – the green squares make up the first diagonal quarter round, blue the second, gold the third, and grey the fourth. Once the diagonal round has been completed, the state has been through one full double round – a column round, and a diagonal round. The complete state at the end of the double round can be seen below:

5	A	6	F
B	7	0	9
4	7	1	A
6	1	C	3

These states were calculated using the attached python program ‘ChaCha20.py’ – the mathematical operations were performed in base 10 and converted after the program’s completion, for the sake of coding simplicity.

Q.3

The extended Euclidean algorithm is an extension to the Euclidean algorithm, and computes, in addition to the greatest common divisor of integers x and y, also integers a and b such that *ax + by = gcd(x, y)*.[3] My implementation of the algorithm can be seen below in figure 5.

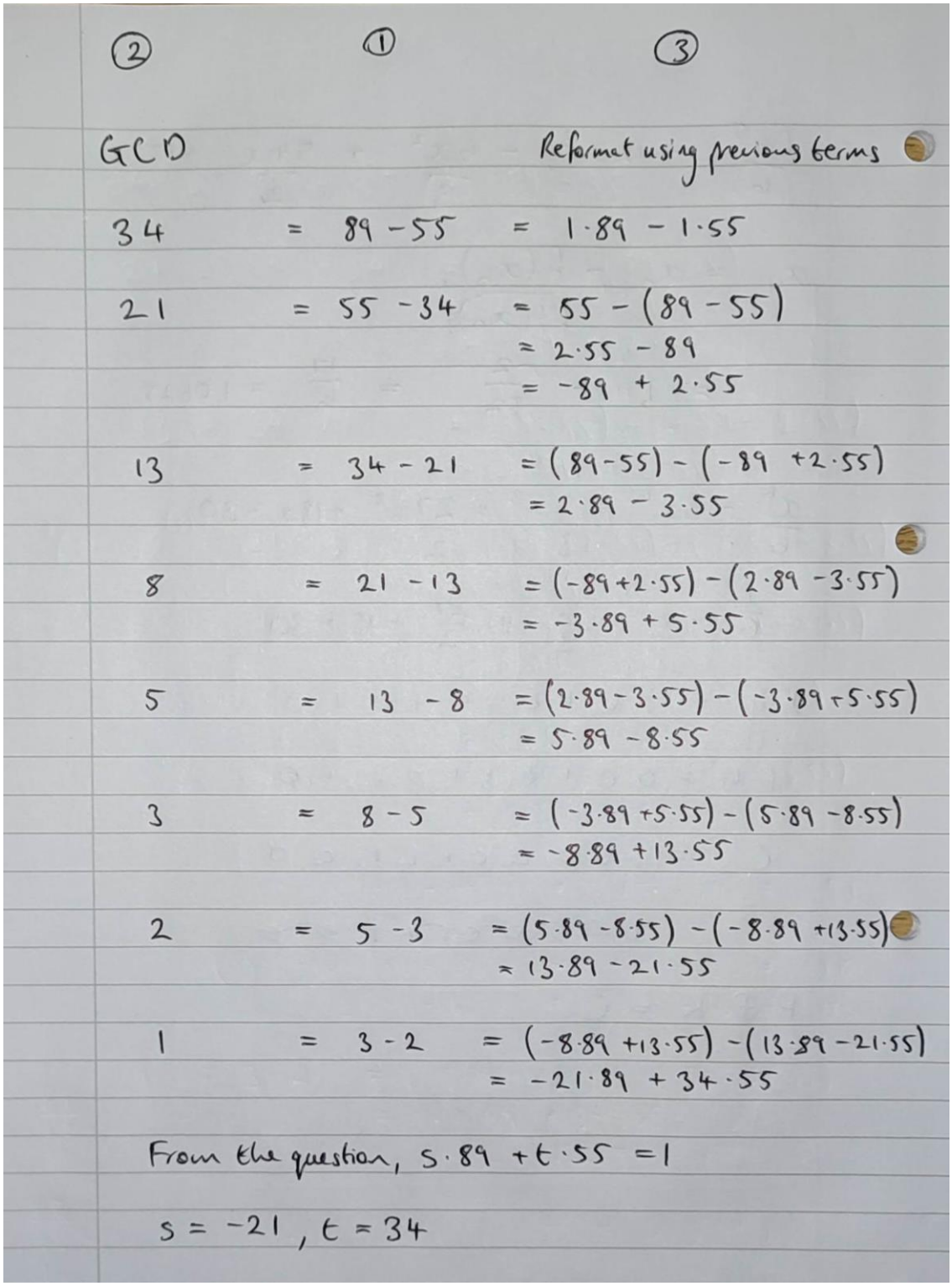

Figure 5: implementation of the extended Euclidean algorithm

Q.4

The multiplicative table for integers modulo 10 can be seen below in figure 6.

x	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	8	0	2	4	6	8
3	0	3	6	9	2	5	8	1	4	7
4	0	4	8	2	6	0	4	8	2	6
5	0	5	0	5	0	5	0	5	0	5
6	0	6	2	8	4	0	6	2	8	4
7	0	7	4	1	8	5	2	9	6	3
8	0	8	6	4	2	0	8	6	4	2
9	0	9	8	7	6	5	4	3	2	1

Figure 6: the multiplicative table for integers mod 10

We know that X has a multiplicative inverse Y if XY = 1. Therefore, we can look at the table and check for two numbers which multiply to give 1 – these two numbers will be each other's inverses.

By the table, the numbers {1, 3, 7, 9} have multiplicative inverses (mod 10).

In general, for the integers modulo N, y has a multiplicative inverse iff y and N are coprime. We can prove this in two directions.

First, assume y and N are coprime.

- If y and N are coprime, their greatest common divisor must be 1.
- Since gcd(y, N) = 1, then there exists some 's' and 't' such that (s * y) + (t * N) = 1, by the Extended Euclidean algorithm.
- We can rework this statement to be (s*y) = 1 – (t * N), and generalise to (s*y) = 1 (mod N)
- Therefore, y has a multiplicative inverse 's' for integers mod N, as their product gives 1.

Next, assume y has a multiplicative inverse 's' for integers mod N.

- We can write this as (s*y) = 1 (mod N)
- Therefore, there exists some 't' such that (s*y) = 1 - (t*N) is satisfied
- Rewrite as (s*y) + (t*N) = 1
    - Assume that y and N have a common divisor > 1, 'd'.
    - Then d divides both y and N
    - Therefore, d divides (s*y) and (t*N), and (s*y) + (t * N)
    - Since (s*y) + (t*N) = 1, d must also divide 1
    - However, the only integers that divide 1 to give an integer are {1, -1}
    - CONTRADICTION
- We then know that the gcd(y, N) is 1, as the gcd of any two integers is >= 1
- Therefore, y and N are coprime as their gcd is 1.

Thus, for integers modulo N, y has a multiplicative inverse iff y and N are coprime.

Q.5

The addition and multiplication tables of F_3^2 formed by polynomials of degree less than two, with coefficients that are integers mod 3, with irreducible polynomial x^2+1, can be seen below in figures 7 and 8 respectively. Figure 9 also shows my working for the result of x * x and (x+1) * (x + 2), to demonstrate my ability to multiply and divide polynomials.

+	0	1	2	x	x + 1	x + 2	2x	2x + 1	2x + 2
0	0	1	2	x	x + 1	x + 2	2x	2x + 1	2x + 2
1	1	2	0	x + 1	x + 2	2x + 1	2x + 1	2x + 2	2x
2	2	0	1	x + 2	x	x + 1	2x + 2	2x	2x + 1
x	x	x + 1	x + 2	2x	2x + 1	2x + 2	0	1	2
x + 1	x+1	x + 2	x	2x + 1	2x + 2	2x	1	2	0
x+ 2	x+2	x	x + 1	2x + 2	2x	2x + 1	2	0	1
2x	2x	2x + 1	2x + 2	0	1	2	x	x + 1	x + 2
2x + 1	2x + 1	2x + 2	2x	1	2	0	x + 1	x + 2	x
2x + 2	2x + 2	2x	2x + 1	2	0	1	x + 2	x	x + 1

Figure 7: addition table for x^2 + 1

.x^2 +1	0	1	2	x	x + 1	x +2	2x	2x +1	2x + 2
0	0	0	0	0	0	0	0	0	0
1	0	1	2	x	x + 1	x + 2	2x	2x + 1	2x + 2
2	0	2	1	2x	2x + 2	2x + 1	x	x + 2	x + 1
x	0	x	2x	2	x + 2	2x + 2	1	x + 1	2x + 1
x + 1	0	x +1	2x + 2	x + 2	2x	1	2x + 1	2	x
x + 2	0	x + 2	2x + 1	2x + 2	1	x	x + 1	2x	2
2x	0	2x	x	1	2x + 1	x + 1	2	2x + 2	x + 2
2x + 1	0	2x + 1	x + 2	x + 1	2	2x	2x + 2	x	1
2x + 2	0	2x + 2	x + 1	2x + 1	x	2	x + 2	1	2x

Figure 8: multiplication table for x^2 + 1

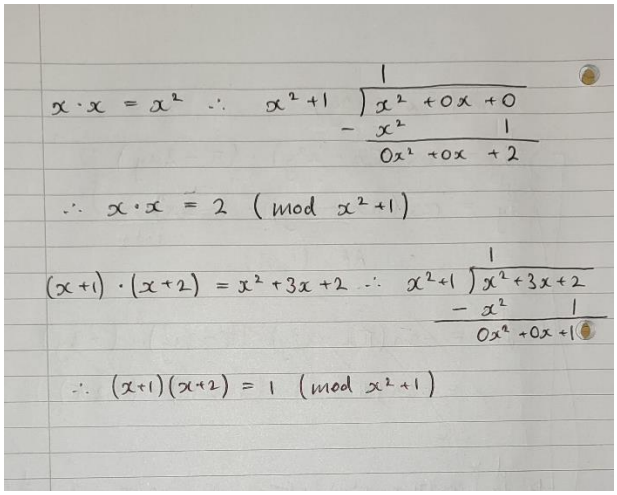

Figure 9: showing working for polynomial multiplication and division

Figures 10 and 11 are the addition and multiplication tables of F_3^2 formed by polynomials of degree less than two, with coefficients that are integers mod 3, with irreducible polynomial $x^2 + x + 2$. Note that the addition tables for irreducible polynomials $x^2 + 1$ and $x^2 + x + 2$ are identical.

+	0	1	2	x	x + 1	x + 2	2x	2x + 1	2x + 2
0	0	1	2	x	x + 1	x + 2	2x	2x + 1	2x + 2
1	1	2	0	x + 1	x + 2	2x + 1	2x + 1	2x + 2	2x
2	2	0	1	x + 2	x	x + 1	2x + 2	2x	2x + 1
x	x	x + 1	x + 2	2x	2x + 1	2x + 2	0	1	2
x + 1	x+1	x + 2	x	2x + 1	2x + 2	2x	1	2	0
x+ 2	x+2	x	x + 1	2x + 2	2x	2x + 1	2	0	1
2x	2x	2x + 1	2x + 2	0	1	2	x	x + 1	x + 2
2x + 1	2x + 1	2x + 2	2x	1	2	0	x + 1	x + 2	x
2x + 2	2x + 2	2x	2x + 1	2	0	1	x + 2	x	x + 1

Figure 10: addition table for x^2 + x + 2

.x^2 +x+2	0	1	2	x	x + 1	x + 2	2x	2x + 1	2x + 2
0	0	0	0	0	0	0	0	0	0
1	0	1	2	x	x + 1	x + 2	2x	2x + 1	2x + 2
2	0	2	1	2x	2x + 2	2x + 1	x	x + 2	x + 1
x	0	x	2x	2x + 1	1	x + 1	x + 2	2x + 2	2
x + 1	0	x + 1	2x + 2	1	x + 2	2x	2	x	2x + 1
x + 2	0	x + 2	2x + 1	x + 1	2x	2	2x + 2	1	x
2x	0	2x	x	x + 2	2	2x + 2	2x + 1	x + 1	1
2x + 1	0	2x + 1	x + 2	2x + 2	x	1	x + 1	2	2x
2x + 2	0	2x + 2	x + 1	2	2x + 1	x	1	2x	x + 2

Figure 11: multiplication table for x^2 + x + 2

In mathematics, an *isomorphism* is a structure-preserving mapping between two structures of the same type that can be reversed by an inverse mapping.[4] In other words, two fields are isomorphic if they are the same after renaming elements. To show that the fields constructed by the two different polynomials are the same, there must be a structure-preserving reversable mapping between the two tables. I do not need to prove this for the addition tables, as they are already identical.

To map an element A from the table constructed with irreducible polynomial $x^2 + 1$ to the table constructed with $x^2 + x + 2$, first subtract the target table's polynomial from the current table's polynomial, i.e., find the difference between them in that direction. Then multiply this result by the product of A's row header and column header 'x' coefficients (mod 3). Finally, add the resulting polynomial to A (mod 3) to give the target table's polynomial in the correct position.

For example, assume we would like to map the result of (x * x) from the field defined by $x^2 + 1$, to the field defined by $x^2 + x + 2$. Firstly, subtract $x^2 + x + 2$ from $x^2 + 1$ to give 2x + 2. Then multiply 2x + 2 by the product of the coefficients of x in the row header and column header of the table – for (x * x), this would be (1*1) = 1. Finally, add (2x + 2) to the result of (x * x) in the current field to attain the result of (x * x) in the field defined by $x^2 + x + 2$. In this case, 2 + (2x + 2) = 2x + 1, which is the correct polynomial in the field defined by $x^2 + x + 2$. This newly attained polynomial in the target field is also in the same position as the result of (x*x) in the original field.

This mapping also works in reverse:

- (x * x) = 2x + 1
- $(x^2 + x + 2) – (x^2 + 1) = (x + 1)$
- (1 * 1) = 1 and 1 * (x + 1) = (x + 1)
- (2x + 1) + (x + 1) = 2, which is correct as this is the number we mapped in the beginning.

In one sentence, you subtract the target field's polynomial from the current field's polynomial, multiply the result by the product of the coefficients of x in the row header and column header of the table, and then add that result to the polynomial in the current field to obtain the polynomial in the target field.

eB = [(ip(A) – ip(B)) * coef(x in eA's header row) * coef(x in eA's header column)]+ eA

(Where e(A) is element of table A in a certain position, e(B) is the element of table B in the same position, ip(B) is the irreducible polynomial which defines the table B, coef(x) is the coefficient of x)

This formula for consistent substitution of polynomials works correctly for all fields that satisfy the requirements given in the specification, not just those that are defined by the irreducible polynomials given in the question. I cannot take the time to exhaustively prove this here, but suppose we take the result of 2x * (2x + 1) in the field defined by $x^2 + x + 2$, and we wish to map it to the field defined by $x^2 + 2x + 2$.

- 2x * (2x + 1) = (x + 1) in the field defined by $x^2 + x + 2$ (taken from figure 11)
- $(x^2 + x + 2) – (x^2 + 2x + 2) = 2x$
- (2 * 2) = 1 (mod 3) and 1 * (2x) = 2x
- 2x + (x + 1) = 1, which is indeed the product of 2x and 2x + 1 in the field defined by $x^2 + 2x + 2$

Therefore, not only are the two fields given in the question isomorphic, (i.e., the same field apart from the naming of the elements), but in fact every field formed by polynomials of degree strictly less than 2, with coefficients that are integers mod 3, defined by an irreducible polynomial, is isomorphic with every other field generated under the same rules.

Parenthetically, the polynomials in the first three rows and columns of the tables / fields are identical after mappings, as the polynomials in their row and column headers (0, 1, 2) have zero as their coefficient of the x term. For example, considering 2 * (x + 1) being mapped from the field defined by $x^2 + x + 2$ to the field defined by $x^2 + 1$:

- 2 * (x + 1) = 2x + 2
- $(x^2 + x + 2) – (x^2 + 1) = (x + 1)$
- 0 * 1 = 0, and so 0 * (x + 1) = 0
- (2x + 2) + 0 = 2x + 2

.x^2 +1	0x + 0	0x + 1	0x + 2	1x	1x + 1	1x +2	2x	2x +1	2x + 2
0x + 0	0	0	0	0	0	0	0	0	0
0x + 1	0	1	2	x	x + 1	x + 2	2x	2x + 1	2x + 2
0x + 2	0	2	1	2x	2x + 2	2x + 1	x	x + 2	x + 1
1x	0	x	2x	2	x + 2	2x + 2	1	x + 1	2x + 1
1x + 1	0	x +1	2x + 2	x + 2	2x	1	2x + 1	2	x
1x + 2	0	x + 2	2x + 1	2x + 2	1	x	x + 1	2x	2
2x	0	2x	x	1	2x + 1	x + 1	2	2x + 2	x + 2
2x + 1	0	2x + 1	x + 2	x + 1	2	2x	2x + 2	x	1
2x + 2	0	2x + 2	x + 1	2x + 1	x	2	x + 2	1	2x

(The x coefficients in the row and column headers have been coloured in red, as there is a rather high likelihood it wasn't clear what I was referring to)

References:

[1] – Lecture 1, slide 14

[2] – Lecture 2, slide 14

[3] – https://en.wikipedia.org/wiki/Extended_Euclidean_algorithm, accessed 05/10/22

[4] - https://en.wikipedia.org/wiki/Isomorphism, accessed 06/10/2022